Product Requirements Document - collabolatte

Author: DA9 **Date:** 2026-01-14

Alignment Note (2026-01-12): This PRD has been explicitly aligned to the Product Brief dated 2026-01-12. MVP scope has been narrowed to a single opt-in programme with minimal notification and no dashboards, programme creation, or conversation-starter features. Prior content that exceeds the brief's MVP boundary has been preserved in the 'Deferred / Post-MVP (from prior PRD)' section for future phases.

Alignment Note (2026-01-14): MVP ops "health" is **ops-only** and gated behind **minimum-N ≥5**. In MVP, Programme Owners/Admins see only: cycle execution health, notification delivery health, trust guardrails status, and operational errors (with correlation IDs). No participation counts, banded adoption signals, or behavioural tracking.

Success Criteria

User Success

Definition of "Worth It":

A participant considers Collabolatte worthwhile when at least one of these occasionally happens, without high cost:

Success Signal	Description
Learning	Saw something new or a familiar problem from a different angle
Connection	Met someone they'd feel comfortable contacting again
Context	Gained insight into another team, client, market, or way of working
Energy	Had a genuine, human break that felt energising rather than draining

Key Insight: It doesn't matter which of these happens—only that at least one occasionally does. The misses are acceptable if the cost stays low and the upside remains plausible.

Proxy (MVP): The system runs monthly without trust complaints or escalations, and occasional unprompted anecdotes of value emerge. Do not add participation metrics or dashboards in MVP.

Business Success

The Minimum Viable Narrative (6-month board story):

> "*This creates value at low cost, without cultural backlash, and is worth continuing or modestly expanding.*"

Evidence Required	Description
Safe operation	Cycles run on the agreed cadence; delivery failures are visible and recoverable

Evidence Required	Description
Voluntary continuity (unmeasured)	People keep opting in or returning without pressure, guilt loops, or metrics
Credible anecdotes	A small number of concrete stories: an introduction made, duplication avoided, work unblocked, or idea sparked
Trust preserved	No negative signals around surveillance, coercion, or time waste

What This Is Not: ROI calculation. The narrative is credibility and optionality, not financial return.

Technical Success

The MVP Bar:

Criterion	Meaning
Reliable delivery	Matches generated and communicated on agreed cadence
Boring predictability	Auditable, unobtrusive, no surprises
No surveillance feel	No sense of monitoring, scoring, or judgement
Graceful failure	Ignoring it for a month has no consequence
Easy exit	Pause, leave, or forget without friction

Success Test: If people keep turning up, nobody feels watched, and a few good stories emerge unprompted, the system is doing its job.

Operational Outcomes (MVP)

In MVP, Collabolatte does **not** provide adoption/participation analytics. We prove safety and reliability with operational evidence only (gated behind **minimum-N ≥5**):

- **Cycle execution health**: last run, next run, succeeded/failed.
- **Notification delivery health**: attempted/sent/failed (no recipient lists), retries exhausted.
- **Trust guardrails status**: minimum-N enforcement, “no individual reporting” invariants, deletion pipeline health (queue size + oldest age).
- **Operational errors**: error category counts for the cycle and correlation IDs for investigation.

Product Scope

MVP – Prove People Will Repeatedly Take the Shot

Philosophy: The thinnest possible slice that tests whether the core loop works. If the MVP can't succeed without a feature, the core hypothesis is probably wrong.

Component	Specification

Component	Specification
Signup	Minimal: name, function/department, location. No mandatory bio.
Programme	One public programme only
Matching	One matching rule (random or lightly constrained). No "better" matching—randomness is a feature, not a limitation.
Notification	Match notification with explicit "first move" prompt
Cadence	Clear, predictable timing
Expected behaviour	Meet or ignore, with no consequence

Explicitly Excluded from MVP:

- Match expiry window (introduces edge cases and policy decisions too early)
- Quiet milestones (nice but not required to validate core loop)
- Multiple algorithms
- Profile customisation beyond basics

Future Vision (post-MVP, in order)

1. **Conversation support** – optional starters and gentle framing for first-time matches.
2. **Safety valves** – one free pass per cycle; match expiry window.
3. **Programme flexibility** – additional programmes with clear purpose; controlled creation by trusted sponsors.
4. **Boundary shaping** – light constraints to encourage cross-function/region matching.
5. **Growth mechanics** – gift a match; time-boxed experiments.

Scope Summary

Tier	Purpose
MVP	Prove people will repeatedly take the shot
Should-Have	Reduce anxiety and social friction
Post-MVP	Expand surface area carefully
Vision	Keep private until behaviour is proven

User Journeys

Journey 1: Tomás – The New Joiner's First Coffee

Opening Scene:

Tomás is three months into his graduate role. He works fully remote from Porto, reporting to a manager in London he's met twice on video. His team is friendly but small—four people, all in his time zone. He knows his immediate work, but the rest of the firm feels like a map with blank spaces.

He receives an email from HR about Collabolatte—"an opt-in way to meet people outside your team." It promises no tracking, no obligation, low stakes. He's curious. He clicks.

Rising Action:

The signup is trivial—his name, department, and location are pre-filled from the corporate directory (SSO integration). He clicks "Join." Done. No bio, no interests, no quiz.

Two weeks later, an email arrives: "You've been matched with **Samira, Principal Consultant, Singapore**." The email includes a simple first-move prompt with suggested copy to help him reach out.

Climax:

They meet on Thursday. Samira is warm, curious, and asks him what it's like to join a firm remotely. Thirty minutes passes quickly. He learns that the firm has a whole sustainability practice he didn't know existed.

Resolution:

After the call, Tomás feels lighter. He didn't learn anything "useful" in a narrow sense—but he now knows a face, a name, a context. He stays opted in for the next cycle. The map has one fewer blank space.

Journey 2: Marcus – The Sceptic Tests the Waters

Opening Scene:

Marcus is a senior engineer, four years at the firm, works hybrid from Manchester. He's seen HR initiatives come and go. When he hears about Collabolatte, his first instinct is suspicion. Is this another way to measure us?

He reads the fine print. Opt-in. No manager visibility. No feedback required. He clicks "Join" almost out of curiosity.

Rising Action:

His first match arrives: **Jess, Legal Counsel, London**. A lawyer? What would they even talk about?

The email includes a simple first-move prompt with suggested copy. Marcus uses it and they schedule a call.

Climax:

The call is... fine. Not transformative. But the conversation starter gave them a thread to pull. They talk about how legal sees engineering requests, about the frustrations of working across departments. It's oddly humanising.

Resolution:

Marcus doesn't report anything. He simply stays opted in. The next month, he gets matched with someone in product—they realise they've been duplicating work and agree to share notes.

Marcus tells a colleague: "It's not a waste of time. Low stakes. Occasionally useful."

Domain-Specific Requirements

Domain Pattern: Trust Infrastructure

Collabolatte operates in the **Organisational Network Design** domain. The core pattern is that this is **trust infrastructure**, not engagement tooling. Every design decision must pass through the lens:

> "Does this preserve or erode participant trust?"

If it erodes trust, it doesn't ship.

Compliance & Regulatory

Requirement	Detail
GDPR (EU)	Lawful basis for processing employee data—likely legitimate interests with transparency, not consent (consent under employment relationship is problematic)
UK GDPR	ICO guidance on worker monitoring applies—must demonstrate transparency and proportionality
CCPA (California)	Employee data in scope as of January 2023—notice and rights obligations
Cross-border transfers	EU SCCs or UK IDTA/Addendum required for multinational deployments
DPIA	Recommended (and likely required) given employee data and potential privacy impact

Key Insight: None of these block the MVP. They require design-time decisions (data minimisation, transparency, transfer mechanisms) rather than feature gates.

Retention Policy: Match history retained for 12 months, then anonymised. Users can request deletion at any time (GDPR Article 17).

Technical Constraints (Privacy-by-Design)

Constraint	Implication
Data minimisation	Capture only what's needed for matching (name, function, location). No profile richness beyond that.
Purpose limitation	Participation data cannot be repurposed for performance evaluation or surveillance
Retention limits	Match history retention should be time-bounded; define retention period

Constraint	Implication
No individual-level reporting	Aggregate metrics only; no manager access to participation data
Audit trail	Log what the system does (match-sent, match-acknowledged), not what users do

Cultural Risk Mitigations

Risk	Mitigation
Ghost Town	"Practice coffee" framing, conversation starters, match-acknowledged signal
Obligation Creep	Hard policy: no manager visibility, no OKR linkage, explicit voluntary framing
Privacy Breach Perception	Transparent architecture, published data model, no hidden logging
Surveillance Rumours	"We waited" copy, graceful dormancy, explicit opt-out without consequence

Anti-Mandate Principle: Programme Owners must never position Collabolatte as required, expected, or tied to performance. Violation of this principle erodes trust and defeats the product's purpose.

Innovation & Novel Patterns

Detected Innovation Areas

1. Category Creation: "Ritual Engine"

Collabolatte is not an employee engagement tool, an internal social network, or a matching algorithm. It is **infrastructure for manufactured serendipity**—a ritual engine that creates excuses for curious people to meet.

This is genuinely novel framing. Existing tools (RandomCoffee, CoffeePals) position as "coffee matching" or "employee connection." Collabolatte positions as **structural intervention for weak-tie formation**. That's a different mental model.

2. Anti-Measurement as Product Philosophy

Most enterprise tools optimise for measurability. Collabolatte explicitly optimises for **unmeasurability**:

- "Opportunity creation, not outcome attribution"
- "Trust over measurement"
- "If learning competes with ease of use, ease of use wins"

This is contrarian in the enterprise software space. It's also aligned with the research (weak-tie value is diffuse and hard to attribute).

3. Graceful Failure as Feature

"The system didn't punish her absence. It waited."

Most engagement tools penalise inactivity (nudges, reminders, guilt). Collabolatte has designed **graceful dormancy** as a first-class feature. The system is patient. That's unusual.

4. Randomness as Design Choice

"No 'better' matching—randomness is a feature, not a limitation."

This challenges the assumption that algorithms should optimise for "good" matches. The argument: **any cross-boundary match is valuable** because the goal is opportunity creation, not outcome optimisation.

Market Context & Competitive Landscape

Category	Examples	Gap Collabolatte Fills
Enterprise social networks	Microsoft Viva Engage, Slack, Workplace	Coordination tools, not connection tools
Lightweight pairing tools	RandomCoffee, CoffeePals	Match and notify, but position as "coffee chat apps"
Employee engagement platforms	Lattice, Culture Amp	Measurement-heavy, survey-driven

Collabolatte's Position: Low-friction, trust-first, measurement-light structural intervention.

Validation Approach

The innovation is primarily **philosophical/framing**, not technological. Validation is behavioural:

Innovation	Validation Signal
Ritual engine framing	Do people stay opted in without being nudged?
Anti-measurement philosophy	Does trust remain high with minimal feedback collection?
Graceful dormancy	Do dormant users return when ready?
Randomness as feature	Do random matches produce comparable value to optimised ones? (Don't need to prove—just not disprove)

Note: "Randomness as feature" is a hypothesis, not a proven design principle. If early feedback suggests poor match quality is driving drop-off, we may introduce light preference-based matching in Growth phase.

Risk Mitigation

Innovation Risk	Fallback
"Ritual engine" framing confuses buyers	Fallback messaging: "Automated coffee matching for cross-team connection" (less ambitious, more familiar)
Anti-measurement fails to produce evidence	Side-channel anecdote capture; executive champion interviews (narrative over data)
Graceful dormancy looks like low engagement	Reframe metric: "return rate after dormancy" as success signal
Randomness produces bad experiences	Add light constraint (cross-boundary) to prevent sibling-team matches

Lightweight SaaS Specific Requirements

Project-Type Overview

Collabolatte is a **Ritual Engine / Lightweight SaaS**—not a traditional B2B platform. The architecture prioritises simplicity, trust, and learning velocity over feature completeness or enterprise scalability. Every technical decision passes through the filter: "Does this help us test the ritual, or does it accidentally build a platform?"

Cost Philosophy: Run the MVP for as little as possible so the only thing being validated is whether people actually meet.

Technology Stack

Layer	Technology	Notes
Frontend	React	SPA hosted on Azure Static Web Apps
Backend	C# Azure Functions (.NET isolated worker)	Integrated with Static Web Apps
Auth	Azure Entra ID (EasyAuth)	Built-in integration, zero custom auth code
Database	Azure Table Storage	Accessed via Azure.Data.Tables SDK
Email	Azure Communication Services	C# SDK
Hosting	Azure Static Web Apps	Free tier

Why This Stack:

- First-class integration between Static Web Apps, Functions, and Entra ID
- Managed auth handles tokens, sessions, claims automatically
- Strong typing with C# backend and typed Table Storage entities
- Local development via SWA CLI + Azure Functions Core Tools
- GitHub Actions deployment built-in

Azure Architecture (Minimal Viable Infrastructure)

Component	Implementation	Cost
Frontend	Azure Static Web Apps (Free tier)	£0
API	Azure Functions (Consumption plan)	£0
Auth	Azure Entra ID via EasyAuth	£0
Database	Azure Table Storage	~£0.01/month
Email	Azure Communication Services	~£2-5/month
Scheduled Jobs	Azure Functions Timer Trigger	Included

Total Services: 3 (Static Web App, Storage Account, Communication Services)

Estimated MVP Cost: ~£5/month

Authentication Architecture

Entra ID Authentication Flow:

1. User visits app → Static Web Apps redirects to Entra ID login
2. User authenticates (SSO if already logged into M365)
3. Entra ID returns token → Static Web Apps sets session cookie
4. React app calls `/.auth/me` to get user claims
5. API calls include auth automatically (EasyAuth forwards claims)
6. C# Functions receive `ClaimsPrincipal` with user email, name, etc.

No custom auth code needed. Configure identity providers in `staticwebapp.config.json`.

Directory Data Strategy:

Phase	Approach
MVP	Use Entra ID claims (email, name). User confirms/adds department and location on first login.
Post-MVP	Microsoft Graph API enrichment (`User.Read` permission) if directory data quality is good

---

Multi-Tenancy Model

Phase	Approach	Rationale
MVP	Single-tenant per organisation	Simplifies trust, compliance, and failure analysis
Future	Multi-tenant with organisation-level isolation	Only once behaviour is proven

---

Permission Model

Role	Capabilities	Notes

Role	Capabilities	Notes
Participant	Join/leave programmes, receive matches and notifications	Nothing else
Programme Owner	Configure cadence and constraints, view aggregate signals	No individual visibility
Admin	SSO/identity config, data retention, operational oversight	Ops only—NO analytics access

Trust Enforcement: Admin has operational access but NOT analytics access. Architectural separation.

Data Model (Azure Table Storage)

Entity	PartitionKey	RowKey	Fields
User	`"users"`	`{email}`	name, function, location, joinedAt
Programme	`"programmes"`	`{programmeId}`	name, cadence, constraints, ownerId, createdAt
Membership	`{userEmail}`	`{programmeId}`	status, joinedAt, pausedAt
Match	`{programmeId}`	`{matchDate}_{matchId}`	user1Email, user2Email, status, sentAt
Event	`{programmeId}`	`{timestamp}_{eventType}`	eventType, userId, metadata

Design Principles:

- Partition keys aligned to query patterns
- Denormalise where necessary (no joins)
- Simple lookups: user by email, users in programme, matches for programme

---

Notification Architecture (Email + Teams Deep Links)

Notification Flow (MVP):

1. Match generated → system sends **email** to both users
2. Email contains:
    - Match name, department, location
    - Conversation starter prompt
    - **[Start Conversation]** button → Teams deep link: `https://teams.microsoft.com/l/chat/0/0?users={matched_person_email}`
3. User clicks → Teams opens with new chat to their match → user types intro
4. System logs: `match-sent`

Trade-offs Accepted:

- User types their intro (no bot-initiated message)
- Can't detect `match-acknowledged` (no bot)

- Acceptable for MVP—retention is the primary signal

Why This Works:

- No Bot Framework complexity
- No IT permissions negotiation
- Email is reliable and universal
- Teams deep link works across desktop, web, mobile

Integration Architecture

Integration	MVP Approach	Notes
SSO	Azure Entra ID via EasyAuth	Built-in
Directory data	Entra ID claims + user self-service	Graph API Post-MVP
Teams	Deep links in email notifications	No bot
Email	Azure Communication Services	Match notifications

Explicitly Not MVP: Teams bot, Slack, Calendar integration, In-app messaging

---

Matching Algorithm

Constraint	Implementation
Cross-boundary	Must match across function OR location (not same team)
No repeat matches	Within rolling window (e.g., 6 months)
Pair formation	Simple random pairing within constraints

Edge Case Handling:

Scenario	Handling
Odd number	One person skipped; log `match-skipped-no-pair`
Small pool (<10)	Warn Programme Owner
Boundary data missing	Fall back to pure random

Privacy-by-Design Enforcement

Constraint	Implementation
No individual-level API endpoints	Don't build them
Aggregate queries minimum group size	5+ participants
Admin ≠ Analytics	Architectural separation

Constraint	Implementation
Published event schema	Document every event logged

Event Logging

Event	Purpose	Visibility
`user-joined-programme`	Participation tracking	Aggregate only
`match-sent`	Delivery confirmation	Aggregate only
`match-skipped-no-pair`	Small pool detection	System ops
`user-left-programme`	Churn signal	Aggregate only

Compliance Architecture

Requirement	Phase	Implementation
GDPR / UK GDPR	MVP (by design)	Legitimate interests; data minimisation
Worker monitoring alignment	MVP (by design)	No behavioural scoring, no hidden tracking
ISO 27001 / SOC 2	Post-MVP	Not required to test hypothesis

Project Scoping & Phased Development

MVP Strategy & Philosophy

MVP Type: Validated Learning MVP

Goal: Test whether people will repeatedly take the shot—not build a complete product.

Philosophy: The smallest possible thing that validates the core hypothesis. If the MVP can't succeed without a feature, the hypothesis is probably wrong.

MVP Feature Set (Phase 1)

Timeline: 4-6 weeks (solo developer) or 2-3 weeks (small team)

Core Capabilities:

Capability	Description	Journey Support
SSO Login	Azure Entra ID authentication	All journeys
User Profile	Name, department, location (self-service if directory data missing)	Tomás, Marcus

Capability	Description	Journey Support
Programme Join	One pre-configured programme, one-click join	Tomás, Marcus
Programme Leave	Opt-out at any time, no friction	Marcus, Aisha
Match Generation	Random + cross-boundary constraint, scheduled	All journeys
Email Notification	Match details + conversation starter + Teams deep link	All journeys
Aggregate Metrics	Retention rate, match count, boundary crossing %	Priya, Helena

User Journeys Supported:

Journey	MVP Support
Tomás (New Joiner)	Partial—user types intro (no one-click send)
Marcus (Sceptic)	Full
Priya (Programme Creator)	Partial—one fixed programme, basic metrics
Helena (Executive Sponsor)	Partial—aggregate metrics only
Aisha (Dormant User)	Full

Explicitly Excluded from MVP:

Feature	Reason
Profile customisation	Not needed to test hypothesis
Multiple programmes	Adds complexity without learning value
Multiple matching algorithms	Randomness is a feature
Teams bot / Adaptive Cards	Too heavy for MVP
Match expiry window	Introduces policy decisions too early
Calendar integration	Coordination is user's responsibility
Quiet milestones	Nice but not essential
HRIS integration	Directory self-service is sufficient
SOC 2 / ISO 27001	Certifications are Post-MVP

Post-MVP Features

Phase 2: Growth (Post-Validation)

Goal: Reduce friction, expand surface area.

Trigger: Multiple safe monthly cycles completed with no trust escalations, and leadership explicitly requests expansion.

Capability	Description
"Practice coffee" framing	First match feels lower-stakes
One free pass per cycle	Explicit permission not to engage
Programme Owner dashboard	Aggregate metrics, boundary analysis
Multiple programmes	Enable different communities
Quiet milestones	"You've had 5 Collabolattes"
Graph API enrichment	Pre-fill department/location from directory
Framing copy refinement	"Most chats will be fine. A few will be useful."

User Journeys Enhanced:

- Tomás – improved onboarding experience
- Priya – full Programme Owner journey with dashboard

Phase 3: Scale (Post-Retention)

Goal: Enterprise readiness, multi-tenant capability.

Trigger: Multiple organisations requesting pilots; sustained qualitative value and continued voluntary participation over time.

Capability	Description
Multi-tenant architecture	Organisation-level isolation
SOC 2 / ISO 27001	Enterprise compliance certification
HRIS integration	Automated boundary constraint validation
Multiple matching algorithms	Interest-based, role-based options
Programme creation by anyone	Self-service programme creation
Teams bot (optional)	One-click intro if IT permits
Ripple connections	"You met X, they know Y"
Coffee roulette mode	Instant match for power users

User Journeys Enhanced:

- Helena – full executive reporting with cross-org view
- New journey: IT Admin (multi-tenant management)

Risk Mitigation Strategy

Technical Risks

Risk	Likelihood	Mitigation
Directory data quality poor	Medium	User self-service on first login; CSV import fallback
Teams deep link doesn't work on mobile	Low	Test on all platforms pre-launch; email is primary
Table Storage query patterns insufficient	Low	Simple patterns; redesign if needed Post-MVP

Market/Adoption Risks

Risk	Likelihood	Mitigation
Ghost Town (matches don't happen)	Medium	Conversation starters; cross-boundary constraint; curated early adopters
Obligation Creep (feels mandatory)	Low	Hard policy: no manager visibility; explicit voluntary framing
Privacy concerns surface	Low	Transparent architecture; published data model; no hidden logging

Resource Risks

Risk	Mitigation
Fewer resources than planned	MVP is already minimal; no further cuts needed
Timeline slips	Single-tenant, no integrations = minimal dependencies
Key person unavailable	Stack is standard; any C#/React dev can continue

Scoping Summary

Phase	Goal	Duration	Trigger
Phase 1: MVP	Validate hypothesis	4-6 weeks	Start now
Phase 2: Growth	Reduce friction	4-8 weeks	>50% retention, 3 cycles
Phase 3: Scale	Enterprise ready	8-12 weeks	Multi-org demand, 6-month retention

Core Scoping Principle: If you're not embarrassed by your MVP, you shipped too late. The goal is learning, not completeness.

MVP Definition of Done

The MVP is complete when:

- ☐ Users can authenticate via Entra ID
- ☐ Users can join the pilot programme
- ☐ Matching runs on schedule (weekly or monthly; **default monthly**; configured as an ops setting, not user-configurable)
- ☐ Match emails are delivered with conversation starter and Teams link
- ☐ At least 20 participants are enrolled
- ☐ First match cycle completes successfully

Functional Requirements

This is the Capability Contract. Every feature traces back to one of these FRs. If it's not here, it doesn't get built.

Reviewed and refined via Party Mode (Winston, Sally, Murat, Mary) - 2026-01-11

---

1. User Identity & Access

ID	Requirement	Phase	Priority	Journey Link
FR1	Users authenticate via Azure Entra ID (EasyAuth); no custom credential storage	MVP	Must	All
FR2	System captures user email and display name from Entra ID claims on first login	MVP	Must	Tomás
FR3	Users can optionally provide department and location if not available from directory	MVP	Should	Tomás
FR4	Users can view their own participation history (programmes joined, past matches)	MVP	Should	Tomás, Priya

2. Programme Participation

ID	Requirement	Phase	Priority	Journey Link
FR5	Users can view programmes they are eligible to join	MVP	Must	Tomás
FR6	Users can join a programme with a single action	MVP	Must	Tomás
FR7	Users can leave a programme at any time; if mid-cycle, pending match is cancelled and match partner is notified	MVP	Must	Tomás
FR8	Users can view programmes they are currently enrolled in	MVP	Must	Tomás
FR9	Users can pause participation in a programme (skip next cycle)	MVP	Should	Tomás

ID	Requirement	Phase	Priority	Journey Link
FR10	Users receive confirmation of programme join/leave actions	MVP	Should	Tomás

3. Matching

ID	Requirement	Phase	Priority	Journey Link
FR11	System executes matching algorithm on a predictable schedule (weekly or monthly; **default monthly**; configured as an ops setting, not user-configurable)	MVP	Must	Marcus (triggered)
FR12	All programme participants are eligible for matching; organisational boundary filtering is a Growth-phase capability	MVP	Must	Marcus
FR13	Random matching with architecture supporting future algorithm configurability	MVP	Must	Marcus
FR14	Matching algorithm avoids repeat pairings within configurable window (default: 3 cycles); historical matches stored per programme	MVP	Must	Marcus
FR15	If participant count is odd, one participant is gracefully excluded with explanation	MVP	Should	Marcus

4. Notifications

ID	Requirement	Phase	Priority	Journey Link
FR16	Match notification email contains: matched participant name, Teams deep link, and a simple first-move prompt with suggested intro copy (copy-paste friendly)	MVP	Must	Marcus
FR17	Email is sent via Azure Communication Services to user's Entra ID email	MVP	Must	Marcus
FR18	Teams deep link opens 1:1 chat with matched participant	MVP	Must	Marcus

7. Privacy & Trust

ID	Requirement	Phase	Priority	Journey Link
FR30	No individual-level API endpoints exist for admin/owner queries about specific participants; individual participants can access their own data via authenticated endpoints	MVP	Must	All

ID	Requirement	Phase	Priority	Journey Link
FR31	No tracking of whether matched participants actually met	MVP	Must	Aisha
FR32	All personal data deletable upon user request (GDPR Article 17)	MVP	Must	All
FR33	No analytics on message content or conversation outcomes	MVP	Must	Aisha
FR34	System logs events, not behaviours (see Event Logging)	MVP	Must	All
FR41	Users can view a "What we collect" transparency page explaining data handling	MVP	Should	All

8. Event Logging

ID	Requirement	Phase	Priority	Journey Link
FR35	System logs programme lifecycle events (created, paused, deactivated)	MVP	Must	Priya, Helena
FR36	System logs matching execution events (run started, completed, participants matched)	MVP	Must	Marcus
FR37	Privacy-preserving aggregate queries only; minimum 5 participants required for any aggregate query	MVP	Must	Helena
FR38	Event log entries are immutable and timestamped	MVP	Must	Helena
FR39	No individual-level event queries exposed to admin/owner roles	MVP	Must	Aisha

Functional Requirements Summary

MVP functional requirements have been aligned to the product brief (single programme, minimal notification, no dashboards or programme creation). Deferred requirements are preserved in the **Deferred / Post-MVP** section below.

Traceability: Every remaining MVP FR links to an MVP-scope journey and success criteria.

Party Mode Refinements Applied:

- FR4: Clarified "own" data access
- FR7: Added mid-cycle leave handling
- FR12: Clarified MVP scope (no boundary filtering yet)
- FR13: Honest about MVP being random-only
- FR16/19: Merged email content specification
- FR20: Added single-reminder constraint
- FR30: Clarified individual vs admin access
- FR37: Added minimum 5-participant threshold
- FR29a: Added admin dashboard requirement (new)

Pre-mortem Additions (2026-01-11):

- FR41: User-facing transparency page
- FR42: Cycle completion notification to Programme Owner
- Anti-Mandate Principle added to Cultural Risk Mitigations

Deferred / Post-MVP (from prior PRD)

Deferred Notification Enhancements

- FR19 Conversation starter selection (post-MVP conversation support)
- FR20 Reminder per match cycle (not in MVP; risks obligation)

These sections were removed from the MVP-aligned PRD but preserved here for future phases.

Journey 3: Priya – The Programme Creator Sets Conditions

Opening Scene:

Priya leads L&D in the UK practice. She's been asked to "do something about cross-team connection" without adding to anyone's workload.

She hears about Collabolatte. The promise: set it up, let it run safely, and avoid measuring people.

Rising Action:

Priya configures the programme: "UK Practice Coffee." She sets the cadence to monthly, match size to 1:1 pairs, and matching rule to random with repeat-avoidance. She writes one line of copy: "Meet someone you wouldn't otherwise. No agenda. No follow-up required."

She shares a simple join link. Volunteers opt in.

Climax:

After a couple of cycles, she checks the ops health: cycles ran, notifications delivered, and there were no trust escalations. She receives an unsolicited message from a senior partner: "I met someone in the Newcastle office I'd never have found otherwise."

Resolution:

Priya doesn't run Collabolatte—she set conditions and stepped back. When Helena asks for a board update, Priya says: "Monthly cycles ran, notification delivery was reliable, no complaints, and a handful of unprompted good stories."

(In MVP, Priya sees ops-only health signals (cycle execution, notification delivery health, trust guardrails status, and operational errors with correlation IDs) gated behind minimum-N ≥5. No participation counts.)

Journey 4: Helena – The Executive Sponsor Defends the Investment

Opening Scene:

Helena is the CHRO. She approved the Collabolatte pilot because the pitch was low-cost, low-risk, and addressed a real problem.

Rising Action:

At the six-month mark, Priya sends a summary:

- The system ran predictably (monthly cycles completed as scheduled)
- Notification delivery remained reliable (failures were visible and fixed quickly)
- Trust posture held (no complaints, no escalations)
- Anecdotes emerged organically (a small handful of concrete stories)

Climax:

At the board meeting, a director asks: "What's the return on this?"

Helena says: "The return is optionality. We've created a low-cost mechanism for cross-team connection that people actually use, that hasn't created backlash, and that occasionally produces real value. It's infrastructure, not a programme."

Resolution:

The board nods. The pilot becomes permanent.

Journey 5: Aisha – The Dormant User Returns

Opening Scene:

Aisha is a manager in Edinburgh, eight years at the firm. She joined Collabolatte early. Her first matches were fine, then forgettable. Then Q4 hit—deadlines, travel, no bandwidth. She ignored three match notifications. The system didn't chase her. No reminder. No "we miss you." Just silence.

By January, Collabolatte had slipped off her radar.

Rising Action:

In March, things slow down. She notices a match notification: **David, Operations Lead, Dublin**.

Aisha is about to lead a similar project. She copies the suggested first-move message and sends it in Teams.

Climax:

David responds within an hour. They meet for 25 minutes. He tells her about the pitfalls—the vendor who overpromised, the comms plan that should have started earlier. Aisha leaves with three concrete actions.

Resolution:

Aisha stays opted in—this time with intent. She tells a colleague: "I ignored it for months. Nothing bad happened. Then it was useful exactly when I needed it."

The system didn't punish her absence. It waited.

(MVP note: do not log "dormancy" or re-engagement as behavioural analytics. Only system events are logged.)

Dormant User Re-engagement Guidance: When a user returns after ≥2 missed cycles, match notification should include subtle acknowledgment: "Good to see you back" or equivalent. No guilt, no metrics—just warmth.

Journey Requirements Summary

Journey	Key Capabilities Revealed
Tomás (New Joiner)	SSO/directory integration, one-click introduction, conversation starter, match-acknowledged logging
Marcus (Sceptic)	Trust architecture, conversation starter (MVP), no manager visibility, graceful mediocrity
Priya (Programme Creator)	Zero-maintenance setup, cross-boundary matching constraint, passive metrics
Helena (Executive Sponsor)	Narrative evidence, aggregate reporting, side-channel anecdote capture
Aisha (Dormant User)	Graceful inactivity, no guilt, re-engagement logging, "We waited" emotional contract

Implicit Requirements (Surfaced by Elicitation)

Requirement	Rationale	Priority
Directory/SSO integration	Pre-fill signup, reduce friction	MVP - hard dependency
One-click introduction action	Reduce first-move friction	MVP
Conversation starter prompt	Tips awkwardness toward usefulness	MVP (promoted from Should-Have)
Cross-boundary matching constraint	Ensures thesis validation	MVP
Delivery confirmation (silent)	Detect notification failures	MVP
Event logging (match-sent, match-acknowledged)	Future-proof for re-engagement metrics	MVP
Re-engagement tracking (silent)	Validates graceful dormancy works	MVP (lightweight)
Optional story capture channel	Surface anecdotes for Helena	Post-MVP (or manual workaround)

Integration Requirements

Requirement	Rationale	Priority

Requirement	Rationale	Priority
SSO/Directory integration	Pre-fill signup, reduce friction, ensure identity	MVP
Teams/Slack integration	Notification delivery and one-click introduction	MVP
HRIS integration	For cross-boundary constraint validation (function, region)	Post-MVP

5. Programme Management

ID	Requirement	Phase	Priority	Journey Link
FR21	Programme Owner can create a new programme with name, description, and cadence	Growth	Must	Priya
FR22	Programme Owner can invite participants by email address or CSV upload	Growth	Should	Priya
FR23	Programme Owner can view aggregate participation metrics (joined, active, paused)	Growth	Should	Priya
FR24	Matching cadence is configurable as an ops setting (weekly or monthly; **default monthly**); no Programme Owner UI for cadence in MVP	MVP	Must	Priya
FR25	Programme Owner can edit programme name and description (cadence is ops-configured in MVP)	MVP	Should	Priya
FR42	Programme Owner receives notification when matching cycle completes, including status and any system errors (no participation counts)	MVP	Should	Priya

6. Administration

ID	Requirement	Phase	Priority	Journey Link
FR26	Admin can view all programmes in the system	MVP	Must	Helena
FR27	Admin can deactivate a programme	MVP	Must	Helena
FR28	Admin can assign Programme Owner role to users	MVP	Must	Helena
FR29	Admin cannot view individual participation or match data	MVP	Must	Helena (trust constraint)
FR29a	Admin can view ops-only programme health (cycle status, notification delivery health, trust guardrails status, operational errors with correlation IDs) gated behind minimum-N ≥5	MVP	Should	Helena

Non-Functional Requirements

Quality attributes for Collabolatte MVP – deliberately boring, defensible, and aligned with the product's low-pressure nature.

Performance

ID	Requirement	Measure	Rationale
NFR1	Page loads complete within 3 seconds on standard corporate network	95th percentile < 3s	Weekly touchpoint, not productivity tool
NFR2	No hard real-time requirements	N/A	Matching runs on schedule, not on-demand
NFR3	System remains usable under degraded conditions	Core flows complete even if slow	Occasional slowness acceptable; failure is not

Security

ID	Requirement	Measure	Rationale
NFR4	All data encrypted in transit (TLS 1.2+)	Certificate validation on all endpoints	Baseline enterprise expectation
NFR5	All data encrypted at rest	Azure Storage Service Encryption enabled	Defence in depth
NFR6	Authentication via Azure Entra ID only; no custom credential storage	Zero stored passwords	Reduces attack surface to zero
NFR7	Least-privilege access: users see only their own data; owners see only aggregate	Role-based access enforced at API layer	Trust architecture is the product
NFR8	Clear data retention policy: personal data deleted within 30 days of account deletion request	GDPR Article 17 compliance	Right to erasure is non-negotiable
NFR9	No dark telemetry or behavioural tracking	Code review confirms no hidden analytics	Anti-surveillance is a feature
NFR10	Tenant isolation: single-tenant MVP with architecture supporting future multi-tenant separation	Logical isolation by design	Avoid painful refactoring later

Availability & Reliability

ID	Requirement	Measure	Rationale

ID	Requirement	Measure	Rationale
NFR11	Best-effort availability; no formal SLA for MVP	Target 99% monthly uptime (informational)	Convenience service, not critical system
NFR12	One hour of downtime during business hours is acceptable	No pager duty for MVP	Right-sized for internal pilot
NFR13	Clear failure behaviour: no silent errors, no partial matches	All failures logged and surfaced	Users must trust the system told them the truth
NFR14	Matching algorithm is atomic: runs completely or not at all	No half-matched cohorts	Consistency over speed

Email Delivery

ID	Requirement	Measure	Rationale
NFR15	Match notifications sent within 60 minutes of algorithm completion	95% delivered within 60 mins	Reliability over speed
NFR16	Email delivery includes retry logic for transient failures	3 retries with exponential backoff	Azure Communication Services handles this
NFR17	Failed email delivery logged for operational review	Admin-visible delivery failures	Debuggability matters

Accessibility

ID	Requirement	Measure	Rationale
NFR18	WCAG 2.1 AA as aspiration, not hard gate	Best-effort conformance	Good practice without blocking delivery
NFR19	Keyboard navigation for all core flows	Tab order works; no mouse-only interactions	Low-hanging fruit
NFR20	Readable contrast ratios (4.5:1 minimum for text)	Automated contrast checking in build	Easy to enforce
NFR21	Screen-reader friendliness where straightforward	Semantic HTML, ARIA labels on interactive elements	Falls out of good design

Scalability

ID	Requirement	Measure	Rationale
NFR22	Architecture supports future multi-tenant expansion	Partition key strategy accommodates org isolation	Don't paint yourself into a corner

ID	Requirement	Measure	Rationale
NFR23	No requirement to prove scale in MVP	Single-tenant pilot is sufficient	Validate hypothesis first
NFR24	Consumption-based infrastructure (Azure Functions, Table Storage)	No fixed capacity provisioning	Scales down to near-zero; scales up if needed

Observability

ID	Requirement	Measure	Rationale
NFR25	Sufficient logging to debug failures and explain what happened	Structured logs for all API calls and background jobs	Operational necessity
NFR26	No user-level behavioural analytics beyond operational necessity	No tracking of user actions beyond system events	Anti-surveillance by design
NFR27	Application Insights or equivalent for error monitoring	Alerts on elevated error rates	Know when things break

Non-Functional Requirements Summary

Category	NFR Count	MVP Hard Gate	Aspiration
Performance	3	2	1
Security	7	7	0
Availability & Reliability	4	3	1
Email Delivery	3	3	0
Accessibility	4	0	4
Scalability	3	1	2
Observability	3	3	0
Total	**27**	**19**	**8**

Design Principle: These NFRs are deliberately boring. Security is non-negotiable. Everything else is right-sized for a trust-first internal pilot where the goal is learning, not perfection.

Document Summary

What This PRD Defines

Collabolatte is a **ritual engine for manufactured serendipity**—infrastructure that creates excuses for curious people to meet across organisational boundaries.

Dimension	Summary

Dimension	Summary
Product Type	Lightweight SaaS / Ritual Engine
Target Users	Employees in large multinationals seeking cross-team connection
Core Hypothesis	If we make it easy and low-stakes, people will repeatedly take the shot
MVP Scope	Aligned to product brief; see Deferred section for removed items
Technical Stack	Azure Static Web Apps, C# Functions, Table Storage, Entra ID
Estimated Cost	~£5/month
Timeline	4-6 weeks (MVP)

Traceability Chain

```
Vision → Success Criteria → User Journeys (2) → Functional Requirements (MVP-aligned) → Non-Functional Requirements (27)
```

Every requirement traces back to a user journey. Every journey traces back to success criteria. The chain is complete.

Next Steps

Artefact	Purpose
UX Design Document	Interaction flows, wireframes, component specifications
Architecture Document	Technical design, API contracts, data flow diagrams
Epic Breakdown	Development work packages with acceptance criteria

---

PRD Complete. Ready for downstream consumption.